

METHOD AND SYSTEM FOR PRESENTING SYSTEM VIEW FOR AN ANDROID APPLICATION

Field of Invention

5 The invention generally relates to Android system with multiple contexts and more particularly, method and system for presenting a system view for an Android application in an Android system with multiple contexts.

Background

10 As Bring Your Own Device (BYOD) scheme becomes popular in enterprise and other collaborative environments, a portable computing device becomes an integrated environment for applications and data from different domains, such as work domain and personal domain. To support multiple domains and prevent privacy leakage from one domain to another, various technologies have been adopted to separate the portable computing device into
15 multiple contexts.

 One existing technology is a virtualization-based solution. This solution supports multiple contexts by duplicating the whole Android system instance or the Android framework. Each context has its own system view. However, this solution incurs heavy performance or storage overhead due to the duplication of
20 the whole Android system instance.

Another existing technology is a container-based solution. This solution supports multiple contexts by dividing the Android system into multiple containers, which requires repackaging and re-signing of Android applications. The repackaged application is recognized as a new Android application and
5 assigned with a new UID (User Identification). Therefore, this solution breaks the existing application signing strategy and may cause compatibility problems. Moreover, the maintenance cost for dealing with version updates of the Android system is very high.

Another existing technology is Multiple Users per Device solution, i.e., one
10 device supports multiple users and allows explicitly switching among different user environments, i.e., contexts, in the device. However, this solution does not support multiple user environments running simultaneously and has limitations on flexibility since all of the applications running on the current active user environment have to be suspended when the whole device is switched into
15 another user environment.

Summary of Invention

Embodiments of the invention provide a low performance and storage overhead but high flexibility and compatibility solution for presenting different system views for different contexts in a portable computing device.

20 Each context is a separate environment to isolate Android applications and data within the Android system. The context can be identified by the system configuration of the Android system or from sensor information, e.g., Global

Positioning System (GPS). Each Android application is associated with a context, i.e., a current context of the Android application. The context switch for one Android application refers to switching the associated/current context of the Android application.

5 According to a first aspect of the invention, a system for presenting a system view for an Android application in an Android system is provided, where the system comprises:

 a context management module configured to maintain at least one first mapping, wherein each of the at least one first mapping is between an Android
10 application and its current context,

 a policy management module configured to maintain at least one second mapping, wherein each of the at least one second mapping is between a system view and a policy set for defining the system view; and

 a resource access monitor module configured to maintain at least one
15 third mapping, wherein each of the at least one third mapping is between a system view and a combination of an Android application and a context, obtain a resource access request from a requesting Android application, obtain a current context of the requesting Android application according to the at least one first mapping maintained by the context management module, ascertain a current
20 system view to be presented for the requesting Android application based on a third mapping between a system view and the combination of the requesting Android application and the obtained current context of the requesting Android

application maintained by the resource access monitor module, obtain a policy set corresponding to the ascertained current system view according to the second mapping maintained by the policy management module, and present the ascertained system view to the requesting Android application according to the
5 obtained policy set.

With reference to the first aspect, in a first possible implementation manner of the first aspect, the context management module is further configured to receive a context switch request for switching a current context of a target Android application to a target context, update the current context of the target
10 Android application as the target context.

With reference to the first possible implementation manner of the first aspect, in a second possible implementation manner of the first aspect, the context management module is further configured to request the resource access monitor module to lock a resource access of the target Android application after receiving the context switch request and before updating the
15 current context of the target Android application; and request the resource access monitor module to unlock the resource access of the target Android application after updating the current context of the target Android application.

With reference to the second possible implementation manner of the first
20 aspect, in a third possible implementation manner of the first aspect, the resource access monitor module is further configured to lock the resource access of the target Android application upon receiving a request from the context management module to lock the resource access of the target Android

application; and unlock the resource access of the target Android application upon receiving a request from the context management module to unlock the resource access of the target Android application.

With reference to the second or the third possible implementation manner of the first aspect, in a fourth possible implementation manner of the first aspect, the context management module is further configured to after updating the current context of the target Android application as the target context and before requesting the resource access monitor module to unlock the resource access of the target Android application, check whether the target Android application is running on the Android system, and request the Android system to either restart or notify the context switch event to a running process of the target Android application if the Android application is determined as being running on the Android system.

With reference to the fourth possible implementation manner of the first aspect, in a fifth possible implementation manner of the first aspect, the context management module is further configured to request the Android system to terminate, and subsequently restart the running process of the target Android application, wherein the Android system assigns the restarted target Android application a same UID (User ID) which was previously assigned to the target Android application.

With reference to the fourth possible implementation manner of the first aspect, in a sixth possible implementation manner of the first aspect, the context management module is further configured to request the Android system to

notify the context switch event to the running process of the target Android application by notifying the Android system the context switch event for the target Android application.

With reference to any one of the first to sixth possible implementation manner of the first aspect, in a seventh possible implementation manner of the first aspect, the context management module is further configured to receive the context switch request which is triggered by an application communication Intent from a caller Android application to the target Android application.

With reference to the seventh possible implementation manner of the first aspect, in a eighth possible implementation manner of the first aspect, the resource access monitor module is further configured to obtain an application communication Intent from the caller Android application to the Android system; associate a current context of the caller Android application with the Intent; obtain the current context of the target Android application from the context management module and determine whether the current context of the target Android application is the same as the context associated with the Intent, and either deny the Intent or send a context switch request to the context management module for switching the current context of the target Android application to the current context associated with the Intent if the current context of the target Android application is not the same as the current context associated with the Intent.

With reference to any one of the first to sixth possible implementation manner of the first aspect, in a ninth possible implementation manner of the first

aspect, the context management module is further configured to receive the context switch request input by a user of the Android system through a user interface.

With reference to any one of the first to sixth possible implementation manner of the first aspect, in a tenth possible implementation manner of the first aspect, the context management module is further configured to receive the context switch request triggered by predetermined events based on information from a sensor in the Android system.

With reference to the tenth possible implementation manner of the first aspect, in an eleventh possible implementation manner of the first aspect, the sensor includes a GPS, Bluetooth, Radio Frequency Identification (RFID) and/or WIFI.

With reference to any one of the first to eleventh possible implementation manner of the first aspect, in a twelfth possible implementation manner of the first aspect, the target Android application is the requesting Android application.

With reference to any one of the first to twelfth possible implementation manner of the first aspect, in a thirteenth possible implementation manner of the first aspect, the resource access monitor module is a file system monitor module which is further configured to maintain at least one third mapping, wherein each of the at least one third mapping is between a system view for a low level file system in the Android system and a combination of an Android application and a context, obtain a file system access request from the requesting Android

application, determine the current system view to be presented for the requesting Android application based on a third mapping between the system view for the low level file system and the combination of the requesting Android application and the current context of the requesting Android application, obtain
5 a policy set corresponding to the ascertained system view from the policy management module, and apply a runtime monitoring on a file system access to make sure that the requesting Android application views and/or accesses the low level file system according to the policy set.

According to a second aspect of the invention, a method for presenting a
10 system view for an Android application in an Android system is provided, where the method comprises:

obtaining, by a resource access monitor module, a resource access request from a requesting Android application,

obtaining, by the resource access monitor module, a current context of the
15 requesting Android application from a context management module which maintains at least one first mapping, wherein each of the at least one first mapping is between an Android application and its current context,

ascertaining, by the resource access monitor module, a current system view to be presented for the requesting Android application based on a third
20 mapping between a combination of the requesting Android application and the obtained current context of the requesting Android application, and a system view maintained by the resource access monitor module,

obtaining, by the resource access monitor module, a policy set corresponding to the ascertained current system view from a policy management module which maintains at least one second mapping, wherein each of the at least one second mapping is between a system view and a policy set for defining
5 the system view, and

presenting, by the resource access monitor module, the ascertained current system view to the requesting Android application according to the obtained policy set.

With reference to the second aspect, in a first possible implementation
10 manner of the second aspect, the method further comprises: receiving, by the context management module, a context switch request for switching a current context of a target Android application to a target context, updating the current context of the target Android application as the target context.

With reference to the first possible implementation manner of the second
15 aspect, in a second possible implementation manner of the second aspect, the method further comprises: requesting, by the context management module, the resource access monitor module to lock a resource access of the target Android application after receiving the context switch request and before updating the current context of the target Android application; and requesting, by the context
20 management module, the resource access monitor module to unlock the resource access of the target Android application after updating the current context of the target Android application.

With reference to the second possible implementation manner of the second aspect, in a third possible implementation manner of the second aspect, the method further comprises: locking, by the resource access monitor module, the resource access of the target Android application upon receiving a request
5 from the context management module to lock the resource access of the target Android application; and unlocking, by the resource access monitor module, the resource access of the target Android application upon receiving a request from the context management module to unlock the resource access of the target Android application.

10 With reference to the second or the third possible implementation manner of the second aspect, in a fourth possible implementation manner of the second aspect, after updating the current context of the target Android application as the target context and before requesting the resource access monitor module to unlock the resource access of the target Android application, the method further
15 comprises: checking, by the context management module, whether the target Android application is running on the Android system, and requesting the Android system to either restart or notify the context switch event to a running process of the target Android application if the target Android application is determined as being running on the Android system.

20 With reference to the fourth possible implementation manner of the second aspect, in a fifth possible implementation manner of the second aspect, the step of requesting the Android system to either restart or notify the context switch event to the running process of the target Android application comprises:

requesting, by the context management module, the Android system to terminate, and subsequently restart the running process of the target Android application, wherein the Android system assigns the restarted target Android application a same UID (User ID) which was previously assigned to the target
5 Android application.

With reference to the fourth possible implementation manner of the second aspect, in a sixth possible implementation manner of the second aspect, the step of requesting the Android system to either restart or notify the context switch event to a running process of the target Android application, comprises:
10 requesting the Android system to notify the context switch event to the running process of the target Android application by notifying, by the context management module, the Android system the context switch event for the target Android application.

With reference to any one of the first to sixth possible implementation
15 manner of the first aspect, in a seventh possible implementation manner of the first aspect, the step of the receiving a context switch request comprises: receiving, by the context management module, the context switch request from the resource access monitor module, which is triggered by an application communication Intent from a caller Android application to the target Android
20 application.

With reference to the seventh possible implementation manner of the first aspect, in a eighth possible implementation manner of the first aspect, the method further comprises: obtaining, by the resource access monitor module, an

application communication Intent from the caller Android application to the Android system; associating, by the resource access monitor module, a current context of the caller Android application with the Intent; obtaining, by the resource access monitor module, the current context of the target Android application from the context management module, determining, by the resource access monitor module, whether the current context of the target Android application is same as the current context associated with the Intent, and either denying the Intent or sending a context switch request to the context management module for switching the current context of the target Android application to the current context associated with the intent if the current context of the target Android application is not same as the current context associated with the intent.

With reference to any one of the first to sixth possible implementation manner of the second aspect, in a ninth possible implementation manner of the second aspect, the step of the receiving a context switch request comprises: receiving, by the context management module, the context switch request input by a user of the Android system through a user interface.

With reference to any one of the first to sixth possible implementation manner of the second aspect, in a tenth possible implementation manner of the second aspect, the step of the receiving a context switch request comprises: receiving, by the context management module, the context switch request triggered by predetermined events based on information from a sensor in the Android system.

With reference to the tenth possible implementation manner of the second aspect, in an eleventh possible implementation manner of the second aspect, the sensor includes a GPS, Bluetooth, RFID and/or WIFI.

With reference to any one of the first to eleventh possible implementation
5 manner of the second aspect, in a twelfth possible implementation manner of the second aspect, the target Android application is the requesting Android application.

With reference to any one of the first to twelfth possible implementation
manner of the second aspect, in a thirteenth possible implementation manner of
10 the second aspect, the step of obtaining a resource access request comprises:
obtaining a file system access request from the requesting Android application,

wherein the step of determining a current system view to be presented for
the requesting Android application, further comprises:

determining the current system view to be presented for the requesting
15 Android application based on a mapping between the system view for low level
file system and the combination of the requesting Android application and the
current context of the requesting Android application; obtaining a policy set
corresponding to the ascertained system view from the policy management
module; and applying a runtime monitoring on a file system access to make sure
20 that the requesting Android application views and accesses the low level file
system according to the policy set.

Brief Description of the Drawings

The invention will be described in detail with reference to the accompanying drawings, in which:

Figure 1(a) is a block diagram of a system for presenting a system view according to one embodiment of the invention;

5 Figure 1(b) shows an example of the mappings maintained by the context management module of the system in Figure 1(a);

Figure 1(c) shows an example of the mappings maintained by the policy management module of the system in Figure 1(a);

10 Figure 1(d) shows an example of the mappings maintained by the resource access monitor module of the system in Figure 1(a);

Figure 2(a) is a block diagram showing an example of the system in Figure 1(a);

Figure 2(b) shows an example illustrating the different file system views for one Android application running in different contexts;

15 Figure 3 is a flow chart illustrating the method for presenting a system view for an Android application according to one embodiment of the invention;

Figure 4 is a flow chart illustrating a process of context switch according to one embodiment of the invention;

20 Figure 5 (a) shows a flow chart illustrating a first example of the process for requesting the Android system to either restart or notify the context switch event to the running process of the target Android application in Figure 4;

Figure 5 (b) shows a flow chart illustrating a second example of the process for requesting the Android system to either restart or notify the context switch event to the running process of the target Android application in Figure 4;

Figure 6 is a flow chart illustrating a process for triggering a context switch
5 according to one embodiment of the invention.

Detailed Description of Embodiments of the Invention

In the following description, numerous specific details are set forth in order to provide a thorough understanding of various illustrative embodiments of the invention. It will be understood, however, to one skilled in the art, that
10 embodiments of the invention may be practiced without some or all of these specific details. It is understood that the terminology used herein is for the purpose of describing particular embodiments only, and is not intended to limit the scope of the invention. In the drawings, like reference numerals refer to the same or similar functionalities or features throughout the various views.

15 Embodiments of the invention provide a solution for presenting a system view for an Android application based on a combination of the Android application and its current context. With this solution, each Android application has a same UID (User ID) when running in different contexts. Further, this solution requires neither duplicating the whole Android system instance, nor
20 repacking the Android applications.

Figure 1(a) is a block diagram of the system 100 for presenting a system view for an Android application running on an Android system 10 according to

one embodiment of the invention. As shown in Figure 1(a), the system 100 comprises a context management module 110, a policy management module 120 and a resource access monitor module 130.

5 The context management module 110 is configured to maintain at least one mapping, wherein each of the at least one mapping is between an Android application and its current context. Each Android application corresponds to one current context since one Android application can only run in one context at one time. Each context is a separate environment to isolate Android applications and data within the Android system 10. The context can be identified by the system
10 configuration of the Android system 10 or from sensor information, e.g., Global Positioning System (GPS).

Figure 1(b) shows an example of the mappings maintained by the context management module 110. In this example, the context management module 110 maintains two mappings, one is between Android application App1 and its
15 current context CON1; the other is between Android application App2 and its current context CON2. The current context of each Android application may be updated by the context management module 110 once the context management module 110 receives a context switch request. This will be explained in detail later.

20 The policy management module 120 is configured to define a plurality of system views. Each system view corresponds to a policy set including at least one policy for defining what the system view looks like.

In one possible implementation way, the policy management module 120 may maintain at least one mapping, wherein each of the at least one mapping is between a system view and a policy set for defining the system view. Each system view corresponds to one policy set. Figure 1(d) shows an example of the mappings maintained by the policy management module 120. In this example, the system includes four different system views V1, V2, V3, V4, which are respectively defined by the policy sets P1, P2, P3, and P4. Each policy set P1, P2, P3, or P4 may include at least one policy for regulating how the relevant system view should be presented.

The resource access monitor module 130 is configured to maintain at least one mapping, wherein each of the at least one mapping is between a system view and a combination of an Android application and a context, i.e., <App, Context>. That is to say, each combination of an Android application and a context corresponds to a specific system view. The resource access monitor module 130 is also configured to obtain a resource access request from an Android application 140, obtain a current context of the Android application 140 from the context management module 110, determine a current system view to be presented for the Android application 140 based on the combination of the Android application 140 and the obtained current context of the Android application 140, obtain a policy set for defining the determined system view from the policy management module 120, and present the system view to the Android application 140 according to the obtained policy set.

Typically, the resource access request from an Android application to the resources 150 of the Android system 10 is intercepted by the resource access monitor module 130. Then, the resource access monitor module 130 obtains current context of the Android application from the context management module 110 and determines the system view to be presented for the Android application based on the combination of the Android application and the current context according to the mapping maintained therein. Figure 1(c) shows an example of the mappings maintained by the resource access monitor module 130. According to the mappings shown in Figure 1(d), the resource access monitor module 130 will present a system view V1 to APP1 if APP1 is currently under the context CON1, present a system view V2 to APP1 if APP1 is currently under the context CON2, present a system view V3 to APP2 if APP2 is currently under the context CON1, or present a system view V4 to APP2 if APP2 is currently under the context CON2.

The resource access request from an Android application 140 to the resources 150 of the Android system 10 may be a request to access any of various Android specific resources, e.g., activity, broadcast receiver, content provider, service, Linux system resources, i.e., file system resources, and location resources, etc.

The policy set for defining a system view to be presented for an Android application may be determined based on the properties of the specific resources. For example, if the resource access request is a file system access request to access file system resource, the relevant policy set may specify whether one file

is visible or not; or what metadata should be presented, such as permissions and attributes. If the resource access request is a request to access location resource, the relevant policy set may specify the accuracy level, e.g., the GPS 37.422006 can be presented as 37.0. The policies for defining a system view
5 may be classified into two categories: view-related policies for indicating whether one resource, e.g., one file, is included in the view or not, and runtime-related policies for indicating runtime permissions, e.g., read/write permission of files. Accordingly, the resource access monitor module 130 may determine what should be included in a system view based on relevant view-related policies, and
10 enforces security policies on resource access for runtime restriction based on the relevant runtime-related policies.

System 20 shown in Figure 2(a) is an example of the system 10 in Figure 1(a). In this example, the Android application 240 intends to request access to file system resource 250 in the Android system 20; the file system access
15 monitor module 230 is one example of the resource access monitor module 130. The resource access request obtained by the file system access monitor module 230 is a file system access request.

The functions of context management module 210, policy management module 220 and file system access monitor module 230 are similar to those of
20 the context management module 110, policy management module 120 and resource access monitor module 130 and therefore will not be described here.

In the Android design, available storage for an Android application can be divided into internal storage and external storage. The directories assigned in

both internal and external storage are associated with one Android application. Different system views for both internal and external storage can be provided for an Android application. Figure 2(b) shows an example illustrating the different file system views for one Android application running in different contexts. For example, an Android application stores user credentials including username and password, etc., into a file, e.g., credential.xml in its internal storage and downloads resources, e.g., photos and music, etc., into its external storage. In this example, under context 1, the file system access monitor module 230 redirects the access path of this Android application from /data/com.example/ to /data/com.example/context1/, and redirects the access path from /mnt/sdcard/ to /mnt/sdcard/context1/. This redirection is transparent to the Android application. Therefore, when the Android application makes a request to access credential.xml with the path /data/com.example/credential.xml or access downloaded resources with the path /mnt/sdcard/, the request is redirected by the file system access monitor module 230 to /data/com.example/context1/credential.xml or /mnt/sdcard/context1/ respectively. Similarly, after switching into context 2, the request is redirected by the file system access monitor module 230 to /data/com.example/context2/credential.xml or /mnt/sdcard/context2/ respectively.

Since an Android application runs with the same UID in different contexts in embodiments of the invention, in addition to the default Unix-like file system access control, the context information also needs to be bonded with the low level file system. A runtime monitoring on the file system access is required to

be applied for each Android application to ensure that a proper file system view is presented to the Android application under different contexts. For example, each file is associated with a list of contexts, and any Android application running outside the list of contexts of a file is not allowed to access the file.

5 According to one embodiment of the invention, when the Android application 240 intends to access file system resource 250, the file system monitor module 230 may be further configured to maintain at least one mapping, each between a system view for a low level file system in the Android system and a combination of an Android application and a context, obtain a file system
10 access request from the requesting Android application, determine the current system view to be presented for the requesting Android application based on a mapping between the system view for the low level file system and the combination of the requesting Android application and the current context of the requesting Android application, obtain a policy set corresponding to the
15 ascertained system view from the policy management module, and apply a runtime monitoring on a file system access to make sure that the requesting Android application views and/or accesses the low level file system according to the policy set, e.g., only the files associated with the current context of the requesting Android application can be accessed, etc. Thus, the context
20 information is bonded with the low level file system and the Android application to make sure a proper system view is presented to the Android application.

In this example, the policy set for defining a system view may include a unification policy, i.e., providing a unified view of the file system. Accordingly, the

file system monitor module 230 is further configured to provide a unified view of the file system which is effectively a union of multiple directories; or include a confinement policy, i.e., providing a sub-view of the file system. Accordingly, the file system monitor module 230 is further configured to confine the access in the root directory into its subdirectory; or include a redirection policy, i.e., providing a redirected view of the file system. Accordingly, the file system monitor module 230 is further configured to redirect the access from one path to another path.

The policy set support may be implemented through the following steps:

creating separate directories as application data directories in different contexts, using symlinks to link the corresponding directory to an Android application since the internal storage supports Unix-like file system operations, such as symlink operations; creating dummy files to mimic the effect of symlink since the external storage does not support link operations; creating sub-directories and using the file system monitor module 230 to redirect the root access of external storage to the corresponding sub-directory. In addition, the file system monitor module 230 may be configured to ensure that the symlinks cannot be modified or give a different file system view to a newly created symlinks other than the one which has been specified.

Figure 3 is a flow chart illustrating the method for presenting a system view according to one embodiment of the invention. The method is described with reference to the system 100 shown in Figure 1(a).

In **block** 301, the resource access monitor module 130 obtains a resource access request from an Android application 140.

In **block** 302, the resource access monitor module 130 obtains the current context of the Android application 140 from the context management module 110
5 which maintains at least one mapping, wherein each of the at least one mapping is between an Android application and its current context.

In **block** 303, the resource access monitor module 130 ascertains a current system view to be presented for the Android application 140 based on a mapping between a combination of the Android application 140 and the obtained
10 current context of the Android application 140, and a system view maintained by the resource access monitor module 130.

In **block** 304, the resource access monitor module 130 obtains a policy corresponding to the ascertained system view from the policy management module 120 which maintains at least one mapping, wherein each of the at least
15 one mapping is between a system view and a policy set for defining the system view.

In **block** 305, the resource access monitor module 130 presents the ascertained system view to the Android application 140 according to the obtained policy set.

20 Since the Android application running in different contexts is assigned with the same UID in embodiments of the invention, the context switch can be performed flexibly only for a particular Android application instead of switching

the context for the whole Android system. The system for presenting a system view in embodiments of the invention is further configured to perform a process of context switch as described below.

Figure 4 is a flow chart illustrating a process of context switch according to
5 one embodiment of the invention.

In **block 401**, the context management module 110 receives a context switch request for switching current context of a target Android application, e.g., the Android application 140, to a target context.

In **block 402**, the context management module 110 checks whether the
10 current context of the target Android application maintained by itself is the target context, if no, the method proceeds to block 403, if yes, the method proceeds to block 408 to end the context switch process.

In **block 403**, the context management module 110 requests the resource access monitor module 130 to lock the resource access of the target Android
15 application.

In **block 404**, the context management module 110 updates the current context of the target Android application as the target context.

In **block 405**, the context management module 110 checks whether the target Android application is running on the Android system 10, if yes, the
20 method proceeds to the block 406; if no, the method directly proceeds to the block 407.

In **block 406**, the context management module 110 requests the Android system 10 to either restart or notify the context switch event to the running process of the target Android application.

In **block 407**, the context management module 110 requests the resource
5 access monitor module 130 to unlock the resource access of the target Android application after updating the current context of the Android application.

In **block 408**, the context switch process ends.

Figure 5 (a) is a flow chart illustrating a first example of the process for
requesting the Android system to either restart or notify the context switch event
10 to the running process of the target Android application in Figure 4.

In **block 501a**, the context management module 110 requests the Android system 10 to terminate the running process of the target Android application.

In **block 502a**, the context management module 110 requests the Android
system 10 to restart the running process of the target Android application. The
15 Android system assigns the same UID to the target Android application.

Alternatively, in a second example as shown in Figure 5 (b), the Android
system notifies the context switch event to the running process of the target
Android application. In this example, the process for requesting the Android
system to notify the context switch event to the running process of the target
20 Android application comprises the following steps:

In **block** 501b, the context management module 110 notifies the Android system 10 the context switch event for the target Android application.

In **block** 502b, the Android system 10 receives the context switch event and notifies the context switch event to the pre-registered applications for this
5 event.

In **block** 503b, the pre-registered applications run the event handler for context switch event accordingly.

According to the explanation above, embodiments of the invention provide a more user-friendly context switch mechanism that notifies the Android
10 application with the context switch event. Instead of suspending the Android application or the whole context in the existing technologies, the Android application can handle the context switch event itself and customize its own application logic. Therefore, the Android application can switch among different contexts itself with the same UID. This is especially suitable for system built-in
15 applications to handle system services and data.

In embodiments of the invention, the context switch request for the target Android application received by the context management module may be inputted by a user of the Android system through a user interface, or triggered by predetermined events based on information from a sensor on the Android
20 system, the sensor may include a GPS, Bluetooth, RFID and/or WIFI. The context switch request may also be triggered by an Intent from another Android application, e.g., from a white-list of approved Android applications to the target

Android application when the current context of the approved Android application is different from the current context of the target Android application.

Figure 6 is a flow chart illustrating a process of triggering a context switch request according to one embodiment of the invention.

5 In **block** 601, a caller Android application sends an Intent to the Android system 10. The caller Android application may be any Android application running on the Android system 10 other than the target Android application which is the callee Android application.

10 In **block** 602, the resource access monitor module 110 obtains/intercepts the Intent and associates the current context of the caller Android application with the Intent.

In **block** 603, the Android system 10 receives the Intent.

In **block** 604, the Android system 10 resolves the callee Android application, i.e., the target Android application.

15 In **block** 605, the resource access monitor module 130 obtains the current context of the callee Android application and checks whether the obtained current context is same as the context associated with the Intent, if yes, the method proceeds to block 607; if no, the method proceeds to block 606.

20 In **block** 606, the resource access monitor module 130 denies the Intent, and then the method proceeds to end, or the resource access monitor module 130 sends a context switch request to the context management module 110 to

switch the current context of the callee application to the context associated with the Intent, and then the method proceeds to block 607

In **block** 607, the Android system passes the Intent to the callee Android application.

5 As explained above, a cross-context communication between two Android applications running on the Android system either is denied or triggers a context switch of the callee Android application.

 According to the embodiments of the invention, the system view to be presented for an Android application is determined based on both the Android
10 application itself and its current context. Therefore, the method and system provided by embodiments of the invention can support a context-aware strategy where the system presents different system views for different contexts. For example, it can be applied as sensor-aware file system views. When the portable computing device is positioned inside a room which can be identified by GPS,
15 WIFI, Bluetooth, etc., the file system view can be automatically redirected to a remote storage, for example, the remote storage may be an encrypted pre-configured remote storage. The method and system provided by embodiments of the invention can also support an application-specific strategy where the system presents various system views for different Android applications in one
20 context, for example, a fine-grained application-specific access control may be applied in the same context. Some applications can have high privilege to have full read/write access while others can only have limited access. Therefore, the flexibility and compatibility of the method and system provided in embodiments

of the invention are significantly improved compared with the existing technologies.

Further, the system and method according to embodiments of the invention enable a flexible and user-friendly context switch mechanism. The
5 context of an Android application can be flexibly switched without suspending the running application for context switch and changing the context of the whole portable computing device.

It is to be understood that the embodiments and features described above should be considered exemplary and not restrictive. For example, the above-
10 described embodiments may be used in combination with each other. Many other embodiments will be apparent to those skilled in the art from consideration of the specification and practice of the invention. The scope of the invention should, therefore, be determined with reference to the appended claims, along with the full scope of equivalents to which such claims are entitled. Furthermore,
15 certain terminology has been used for the purposes of descriptive clarity, and not to limit the disclosed embodiments of the invention.

Claims

1. A system for presenting a system view for an Android application in an Android system, comprising:

5 a context management module configured to maintain at least one first mapping, wherein each of the at least one first mapping is between an Android application and its current context,

a policy management module configured to maintain at least one second mapping, wherein each of the at least one second mapping is between a system view and a policy set for defining the system view; and

10 a resource access monitor module configured to maintain at least one third mapping, wherein each of the at least one third mapping is between a system view and a combination of an Android application and a context,

obtain a resource access request from a requesting Android application,

15 obtain a current context of the requesting Android application according to at least one first mapping maintained by the context management module,

ascertain a current system view to be presented for the requesting Android application based on a third mapping between a system view and the combination of the requesting Android application and the obtained current context of the requesting Android application maintained by the resource access
20 monitor module,

obtain a policy set corresponding to the ascertained current system view according to the second mapping maintained by the policy management module, and present the ascertained system view to the requesting Android application according to the obtained policy set.

5 2. The system according to claim 1, wherein the context management module is further configured to receive a context switch request for switching a current context of a target Android application to a target context, update the current context of the target Android application as the target context.

10 3. The system according to claim 2, wherein the context management module is further configured to request the resource access monitor module to lock a resource access of the target Android application after receiving the context switch request and before updating the current context of the target Android application; and request the resource access monitor module to unlock the resource access of the target Android application after updating the current
15 context of the target Android application.

4. The system according to claim 3, wherein the resource access monitor module is further configured to lock the resource access of the target Android application upon receiving a request from the context management module to lock the resource access of the target Android application; and unlock the
20 resource access of the target Android application upon receiving a request from the context management module to unlock the resource access of the target Android application.

5. The system according to claim 3 or claim 4, wherein the context management module is further configured to after updating the current context of the target Android application as the target context and before requesting the resource access monitor module to unlock the resource access of the target Android application, check whether the target Android application is running on the Android system, and request the Android system to either restart or notify the context switch event to a running process of the target Android application if the Android application is determined as being running on the Android system.

6. The system according to claim 5, wherein the context management module is further configured to request the Android system to terminate, and subsequently restart the running process of the target Android application, wherein the Android system assigns the restarted target Android application a same UID(User ID) which was previously assigned to the target Android application.

7. The system according to claim 5, wherein the context management module is further configured to request the Android system to notify the context switch event to the running process of the target Android application by notifying the Android system the context switch event for the target Android application.

8. The system according to any one of claim 2 to claim 7, wherein the context management module is further configured to receive the context switch request which is triggered by an application communication Intent from a caller Android application to the target Android application.

9. The system according to claim 8, wherein the resource access monitor module is further configured to obtain an application communication Intent from the caller Android application to the Android system;

associate a current context of the caller Android application with the Intent;

5 obtain the current context of the target Android application from the context management module and determine whether the current context of the target Android application is the same as the context associated with the Intent, and either deny the Intent or send a context switch request to the context management module for switching the current context of the target Android
10 application to the current context associated with the Intent if the current context of the target Android application is not the same as the current context associated with the Intent.

10. The system according to any one of claim 2 to claim 7, wherein the context management module is further configured to receive the context switch request
15 input by a user of the Android system through a user interface.

11. The system according to any one of claim 2 to claim 7, wherein the context management module is further configured to receive the context switch request triggered by predetermined events based on information from a sensor in the Android system.

20 12. The system according to any one of claim 2 to claim 11, wherein the target Android application is the requesting Android application.

13. The system according to any preceding claim, wherein the resource access monitor module is further configured to maintain at least one third mapping, wherein each of the at least one third mapping is between a system view for a low level file system in the Android system and a combination of an Android application and a context,

obtain a file system access request from the requesting Android application, determine the current system view to be presented for the requesting Android application based on a third mapping between the system view for the low level file system and the combination of the requesting Android application and the current context of the requesting Android application,

obtain a policy set corresponding to the ascertained system view from the policy management module, and apply a runtime monitoring on a file system access to make sure that the requesting Android application views and/or accesses the low level file system according to the policy set.

14. A method for presenting a system view for an Android application in an Android system, the method comprising:

obtaining, by a resource access monitor module, a resource access request from a requesting Android application,

obtaining, by the resource access monitor module, a current context of the requesting Android application from a context management module which maintains at least one first mapping, wherein each of the at least one first mapping is between an Android application and its current context,

ascertaining, by the resource access monitor module, a current system view to be presented for the requesting Android application based on a third mapping between a combination of the requesting Android application and the obtained current context of the requesting Android application, and a system
5 view maintained by the resource access monitor module,

obtaining, by the resource access monitor module, a policy set corresponding to the ascertained current system view from a policy management module which maintains at least one second mapping, wherein each of the at least one second mapping is between a system view and a policy set for defining
10 the system view, and

presenting, by the resource access monitor module, the ascertained current system view to the requesting Android application according to the obtained policy set.

15. The method according to claim 14, further comprising:

15 receiving, by the context management module, a context switch request for switching a current context of a target Android application to a target context, updating the current context of the target Android application as the target context.

16. The method according to claim 15, further comprising:

20 requesting, by the context management module, the resource access monitor module to lock a resource access of the target Android application after

receiving the context switch request and before updating the current context of the target Android application; and

requesting, by the context management module, the resource access monitor module to unlock the resource access of the target Android application
5 after updating the current context of the target Android application.

17. The method according to claim 16, further comprising:

locking, by the resource access monitor module, the resource access of the target Android application upon receiving a request from the context management module to lock the resource access of the target Android
10 application; and

unlocking, by the resource access monitor module, the resource access of the target Android application upon receiving a request from the context management module to unlock the resource access of the target Android application.

15 18. The method according to claim 16 or claim 17, further comprising:

after updating the current context of the target Android application as the target context and before requesting the resource access monitor module to unlock the resource access of the target Android application,

checking, by the context management module, whether the target Android
20 application is running on the Android system, and

requesting the Android system to either restart or notify the context switch event to a running process of the target Android application if the target Android application is determined as being running on the Android system.

19. The method according to claim 18, wherein the step of requesting the
5 Android system to either restart or notify the context switch event to the running process of the target Android application, comprises:

requesting, by the context management module, the Android system to terminate, and subsequently restart the running process of the target Android application, wherein the Android system assigns the restarted target Android
10 application a same UID (User ID) which was previously assigned to the target Android application.

20. The method according to claim 18, wherein the step of requesting the Android system to either restart or notify the context switch event to the running process of the target Android application, comprises:

15 requesting the Android system to notify the context switch event to the running process of the target Android application by notifying, by the context management module, the Android system the context switch event for the target Android application.

21. The method according to any one of claim 15 to claim 20, wherein the step of
20 the receiving a context switch request comprises:

receiving, by the context management module, the context switch request from the resource access monitor module, which is triggered by an application

communication Intent from a caller Android application to the target Android application.

22. The method according to claim 21, further comprising:

obtaining, by the resource access monitor module, an application
5 communication Intent from the caller Android application to the Android system;

associating, by the resource access monitor module, a current context of the caller Android application with the Intent;

obtaining, by the resource access monitor module, the current context of the target Android application from the context management module,

10 determining, by the resource access monitor module, whether the current context of the target Android application is same as the current context associated with the Intent, and

either denying the Intent or sending a context switch request to the context management module for switching the current context of the target
15 Android application to the current context associated with the intent if the current context of the target Android application is not same as the current context associated with the intent.

23. The method according to any one of claim 15 to claim 20, wherein the step of the receiving a context switch request comprises:

20 receiving, by the context management module, the context switch request input by a user of the Android system through a user interface.

24. The method according to any one of claim 15 to claim 20, wherein the step of the receiving a context switch request comprises:

receiving, by the context management module, the context switch request triggered by predetermined events based on information from a sensor in the
5 Android system.

25. The method according to any one of claim 15 to claim 25, wherein the target Android application is the requesting Android application.

26. The method according to any preceding claim, wherein the step of obtaining a resource access request comprises: obtaining a file system access request
10 from the requesting Android application,

wherein the step of determining a current system view to be presented for the requesting Android application, further comprises:

determining the current system view to be presented for the requesting Android application based on a third mapping between the system view for low
15 level file system and the combination of the requesting Android application and the current context of the requesting Android application;

obtaining a policy set corresponding to the ascertained system view from the policy management module; and

applying a runtime monitoring on a file system access to make sure that the
20 requesting Android application views and accesses the low level file system according to the policy set.